Developing a testing system for Functional Programming course

VASILY I. SHAYTAN
Saint-Petersburg State University

We report on our experience teaching a Haskell-based functional programming course to over 400 students for four autumn terms. The syllabus was organized around selected material from various sources. Throughout the terms, we emphasized correctness through tests and proofs. The submission architecture was coupled with hand and automatic testing, giving students the possibility to correct mistakes before the deadline. To motivate the students, we complemented the weekly assignments with an informal competition and gave away points in a end course.

Categories and Subject Descriptors: D.1.1 [**Programming Techniques**]: Applicative (Functional) Programming; D.3.2 [**Programming Languages**]: Language Classifications-Applicative (functional) languages; K.3.2 [**Computers and Education**]: Computer and Information Science EducationComputer science education

General Terms: Algorithms, Languages, Reliability

Additional Key Words and Phrases: Haskell, functional programming, testing, monads, education

1. INTRODUCTION

This paper reports on a mandatory Haskell-based functional programming course at the Saint-Petersburg State University. In the first iteration (autumn-winter semester of 2012-2013), there were 32 students enrolled. In the following autumn-winter semester (2016-2017), there were 80 students enrolled. The course ran for 16 weeks with one 90-minute tutorial each week. The weekly homework was graded, but the final grade was primarily determined by the examination. To make the homework more attractive, we coupled it with an informal programming competition. The departmental course description does not prescribe a specific functional language but focuses on functional programming in general. In the previous two years, the course had been based on WinHugs. We have a Haskell Platform but chose Haskell because of its simple syntax, large user community, real-world appeal, variety of textbooks, and availability of Glasgow Haskell Compiler. The one feature we could well have done without is lazy evaluation; in fact, we wondered whether it would get in the way. The course was mandatory for computer science and information systems students. All had learned Java in their first semester. The computer science students had also taken courses on algorithms and data structures, discrete mathematics, and linear algebra. The information systems students had only had a basic calculus course and were taking discrete mathematics in parallel.

2. SYSTEM DESIGN

As a software development server part of the system has been selected PHP language, as it is already the existing version was written system.
To realize the possibility of compiling and testing prislan- of the solution of the server was installed Glasgow Haskell Compiler (GHC) version 7.10.2 - one of the most powerful and developed up to now there tions day compilers functional language Haskell.

3. DESIGN FEATURES OF THE SYSTEM

Consider the main features of which it was decided to provide chit in the system. At the teacher for each lesson there is a certain a set of tasks. After each lecture at the university, the company's website are available on the new tasks of the topic. Tasks are 2 types:

—base, they are the main, the term of their fulfillment 1 week;
—Additional, the period for their implementation are already 2 weeks since they are a little more complicated.

Two ways to simplify scanning tasks was devised:

—The awarding of the automatic right decisions
—Automatic affixing comments.

Here are the main features of the subsystems The awarding decision tions.

(1) The system stores the solution in the table right decisions
(2) The system compares the solutions based on transformations
(3) It was decided to implement the following conversion:
    —Gaps
    —Parentheses (The system compares the student to the correct decision solution up to add and remove brackets)
    —Renaming variables (system compares decision student with the right solution up to a transfer of names variables. But more about that is below.)
    Note: the use of such transformation might The awarding of the opportunity to receive an automatic syntactic cally incorrect problems, but we can do so, because that tasks are automatically counted only once they have passed the tests.
(4) The awarding of delayed.
    We could have set off the right decisions at once, but to the student did not look for the selection decision, we do not have do. Instead, we have added a button to the teaching interface Vatel, when clicked, the system counts solutions, are marked as "Approved".

We describe the process of automatically The awarding.
For each task, if a student decided he sends the decision to the testing system. The task goes through a series of tests, and if all tests were successful, the student receives this information. Next to it, the message "Wait for verification of the teacher." But this system of action with the student code does not stop. Just about dalneshem behavior of the student should not be aware of the system.
Once the system has sent information about the test, the student is exposed to the processing code. This processing includes Renaming himself variables, taking away unnecessary brackets and spaces. Then, we seek to correct this problem in solution table right decisions, which, after the application of this treatment is the same

as the student's decision. And if you suddenly found such coincidence, the system marks the decision as approved. And a teacher with a special button on your page is to count all such decisions approved system.
Here are the main features of the subsystem is replaced affixing chany.

(1) Keep samples and comments to them in the table
(2) Samples are compared in view of the gaps and the brackets, but excluding renaming variables.
(3) Comments can be inserted both automatically and manually
(4) Automatic notes inserted under that line, in which swarm of the sample was found
(5) Hand remarks inserted in the place where the courses dirty, with text notes automatically switches to the new line.

4. DESCRIPTION OF THE SYSTEM INTERFACE

In this part, we will not describe all the features of system testing. We describe only those features that were added while working on a diploma.

4.1 List of correct decisions

On the main page of the teacher, there is a link to "correct decision" in the upper left corner. By clicking on it, we get to the page that lists all the right decisions (fig. 1).

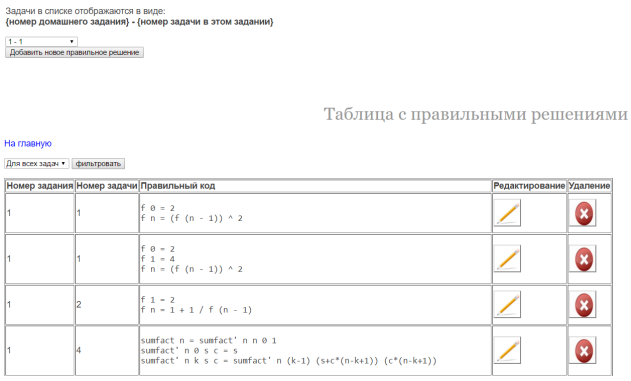

Fig. 1. Table with the right solutions.

The last two columns are responsible for editing and deleting, respectively. When editing, we get to the page where you make changes. When you delete we delete the desired right decision. At the top there is a button "Add new right decision." Clicking on it, we will get to the page to create a new right decision.

4.2 Adding the right decisions

Once we got to add the right solutions page (fig. 2), we have to fill in only the very right decision, as the task number was specified on the previous page. But if you suddenly need to change it, simply enter the desired number of us in the field.

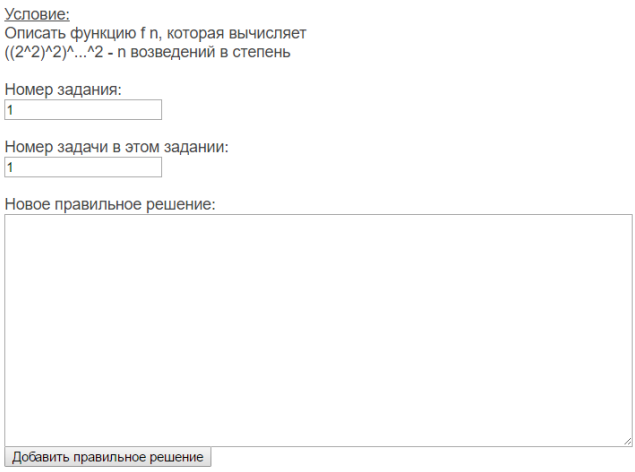

Fig. 2. Adding new right decisions.

4.3 Other features

Editing. On the main page right decisions displays information about all the correct solution for the desired task. It is represented by vvide table, in which the penultimate column is responsible for editing the task. By clicking on the appropriate icon, we get to the edit page right decision (. 3). For change task number or text solutions change the relevant data accept the changes, or they will not be saved.

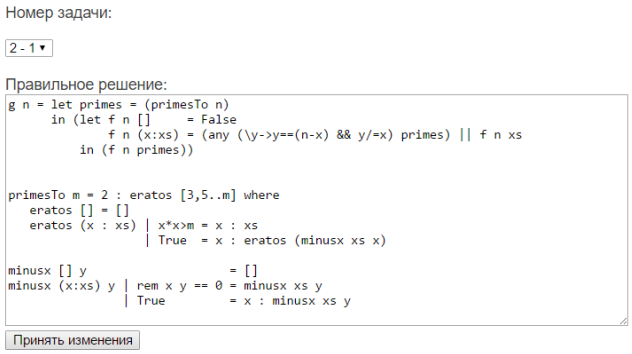

Fig. 3. Editing right decision.

Removal. For each decision on the main page of correct decisions in the last column is the delete icon. When pressed, the system asks you to confirm that we want to remove this particular line. After confirmation, the corresponding row is deleted.

Selection by number problem. Displays all the right decisions a little uncomfortable, so the teacher has the ability to filter by number problem solving. To do this, select the number of the desired task and click "Filter" button.

4.4 The awarding of Automatic

On the main page showing the last teacher sent students solutions (fig. 4). For such solutions have a special filter, in which all the solutions that can be counted automatically labeled as "Approved

system.” After clicking on ”solutions to credit-approved system” all such decisions will automatically be counted.

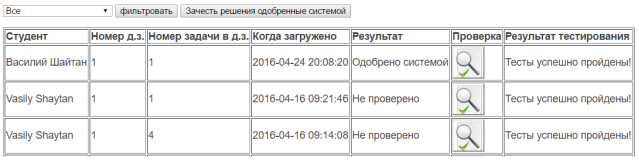

Fig. 4. Decisions adopted by the system.

4.5 List of typical comments

On the teacher’s page in the upper left corner there is a link ”Typical comments”. After clicking on it, we find ourselves on the page where the information is displayed on comments (. 5). Present adding, editing and deleting comments. When you press the right buttons, we get to meet these buttons page.

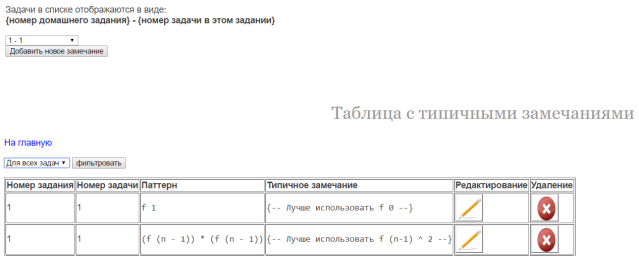

Fig. 5. Page typical comments.

4.6 Adding a new typical comments

Once we got to the addition of a typical page comments (. 6), we need to fill the ”new pattern” and ”New typical remark.” The first field we write the desired pattern, and the second, a remark which should be to bring the system when it is detected. You can change the number of tasks that point to the previous page.

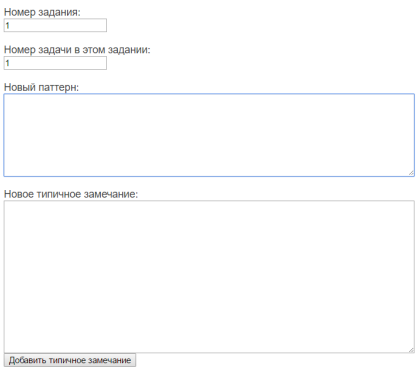

Fig. 6. Adding a typical remark.

4.7 Other features

Editing. On the main page of the typical comments displays information about all the observations for the required task. It is represented by vvide table, in which the penultimate column is responsible for the editing of this remark. By clicking on the appropriate icon, we get to the edit page of the typical comments. To change the sample task number or text notes, change sootvetstvyuschie data. After that, we accept the changes, or they will not be saved. An example of the typical comments edit page is shown in fig. 7

Fig. 7. Editing typical comments.

Removal. For each comment on the main page of the typical comments in the last column is the delete icon. After pressing the system asks you to confirm that we want to remove this particular row from the table. After confirmation of this line is removed.

4.8 Verification task using typical remarks

When checking the problems, the teacher gets to the page where he can comment on the decision, sent by the student. To do this, he has a list of comments for the task to be inserted manually, and a button that allows you to automatically insert comments (fig. 8).

Manually adding typical notes. To insert a comment in the list of comments in the student code, you need to choose a place in the code (usually the end of the line, which is the reason for the comments), where we want it do. Next, select the desired note on the list and click ”Add a comment”. To save written notes, click ”Save Changes.” After that, the student can see his decision with the observations of the teacher, which are highlighted in red for him.

Automatic comments. To automatically add comments to the student code, you need to check the solutions page, click the ”Automatically insert a comment.” Thereafter, all such comments system inserts. If you suddenly need to change the observation that the system put in, go to the field and the decision right there replacing. After not forget to save your changes.

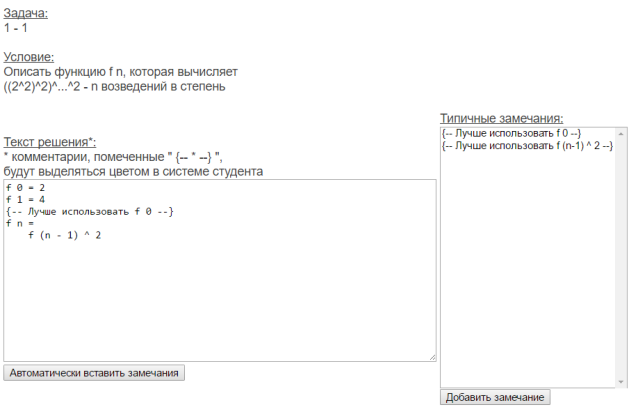

Fig. 8. The interface works with the comments.

5. DESCRIPTION OF THE IMPLEMENTATION

We describe the technical improvements that have been implemented in our system.

5.1 Driving databases

First, it was decided to pereysti sistemamy storage MyISAM to InnoDB [inn 2016]. What are the reasons for the transition?

—InnoDB supports transactions, unlike MyISAM.
—InnoDB supports row-level locking (MyISAM - only table-level).
—InnoDB supports foreign key constraints (MyISAM - no).

When using transactions and foreign keys, database becomes more monolithic, disappear danger phantom records or discontinued operations. A data integrity is important wherever there is a modification of entry / update / delete. Such modifications are so many, as a new student kazhny decision is recorded in the database. Each observation of the teacher, to the left of the task requires line update in the table.
The database table 2 was added.

(1) correctsolutions - table right decisions consists of 3 fields.
—CorrectSolutionID - primary key
—TaskID - task number, for which there is a right decision
—CorrectCode - itself the right decision
(2) tipicalremarks - Table of typical observations consists of 4 fields.
—TypicalRemarkID - primary key
—TaskID - task number for which there is a typical remark
—PatternForTaskID - sample on which we can write a remark
—RemarkText - text of remarks

5.2 Updating outdated functions

Since the system [ms2 2012] was developed in 2011, the PHP programming language has undergone some changes.
MySQL extension date version PHP 5.5.0, and is not recommended for use when writing new code, and will be removed in the future [php 2016]. It was therefore decided to use modern extension PDO (PHP Data Objects)[php 2016]. This extension can support any database management system, for which there is a PDO-driver. For example, PDO_MYSQL, PDO_IBM, PDO_OCI, PDO_DBLIB and many others. Since we have a MySQL database, respectively, we use the driver PDO_MYSQL.

All HTML tags are obsolete have been replaced by new or implemented using CSS.

5.3 renaming variables

Very often you can find solutions that differ only in the names of variables. For such cases, it was implemented a feature that allows you to compare the student's decision and the decision of the table right decisions. System specially renames variables. This can be seen in fig. 9.
All variables are replaced by x1 x2, etc. All the same variable within a single sentence replaced with the same. Also, variables in different sentences are numbered independently, ie identical variables within a single program can have different numbers and conversely. It is worth noting that the system performs all of these activities with the student code only after it has passed all the tests and the compilation.

```
sumfact n = sumfact' n n 0 1
sumfact' n 0 s c = s
sumfact' n k s c = sumfact' n (k-1) (s+c*(n-k+1)) (c*(n-k+1))
```

```
x1 x2 = x3 x2 x2 0 1
x1 x2 0 x3 x4 = x3
x1 x2 x3 x4 x5 = x1 x2 (x3-1) (x4+x5*(x2-x3+1)) (x5*(x2-x3+1))
```

Fig. 9. An example of renaming variables.

5.4 The structure of the project

The project includes the n files with php, 1 file with the extension css and 1 file with the extension js. For completeness of description files is shown in Table I.

6. MODELING THE IRIS DEFORMATION

Testing was conducted teacher. According to test results the following conclusions were reached:ss

—the awarding of a subsystem of automatic tasks works quite efficiently and facilitates verification of simple tasks.
—expressed the wish to add to the subsystem of the automatic solution The awarding different rules of procedure
—subsystem putting comments is also quite convenient and greatly simplifies the job of the teacher
—it expressed the wish to allow the assignment of some observations that are applicable to all tasks
—more complex conditions was automatically check it was desirable than that, such as the use of tail recursion, but the implementation of such checks is a rather complicated task.

The introduction of automatic subsystems The awarding decisions and automatically affixing typical observations planned for the fall semester of 2016-2017 year.

7. CONCLUSION

They have been developed, designed and implemented automated subsystems The awarding of tasks and automatic prostanovka comments. During testing it was demonstrated a significant labor saving teacher.

File	Appointment
authorization.php	user login page
config.php	Provides database connection
footer.php	Information Systems Authors
function.php	List of the basic functions
index.php	Home System page
logout.php	Implements logout
registration.php	User Registration Page
Student work	
allDownloadSolution.php	All downloaded solutions
index.php	Home Student page
loadSolution.php	Loading solutions
profile.php	Page to modify personal data
uploadSolution.php	Output solutions
watchSolution.php	View solutions
Teacher works	
addCorrectSolution.php	Adding the right decision
addTask.php	Adding a task
addTesk.php	Adding test
addTypicalRemark.php	Adding typical comments
checkSolution.php	Page check student solutions
correctSolution.php	Correct solutions
editCorrectSolution.php	Editing right decision
editTest	Editing test
editTypicalRemark	Editing typical comments
hometask.php	Hometasks
index.php	Home teacher page
result	Student results
task.php	Tasks
test.php	Tests
typicalRemark	Home typical comments

Table I. Description of the Project

REFERENCES

2012. Functional programming. (2012). http://ms8255.ru/fp4/

2016. MySQL Reference Manual 7.5 InnoDB Tables. (2016). http://www.mysql.ru/docs/man/InnoDB.html

2016. PHP: Hypertext Preprocessor. (2016). http://php.net/

Received May 2016; accepted June 2016